Aalborg Universitet

AALBORG UNIVERSITY
DENMARK

Network aware dynamic context subscription management

Shawky, Ahmed; Olsen, Rasmus Løvenstein; Pedersen, Jens Myrup; Schwefel, Hans-Peter

Published in:
Computer Networks

DOI (link to publication from Publisher):
10.1016/j.comnet.2013.10.006

Publication date:
2014

Document Version
Publisher's PDF, also known as Version of record

Link to publication from Aalborg University

Citation for published version (APA):
Shawky, A., Olsen, R. L., Pedersen, J. M., & Schwefel, H-P. (2014). Network aware dynamic context subscription management. *Computer Networks*, *58*, 239–253. https://doi.org/10.1016/j.comnet.2013.10.006

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

? Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
? You may not further distribute the material or use it for any profit-making activity or commercial gain
? You may freely distribute the URL identifying the publication in the public portal ?

Take down policy
If you believe that this document breaches copyright please contact us at vbn@aub.aau.dk providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from vbn.aau.dk on: December 27, 2020

Computer Networks 58 (2014) 239–253

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Network aware dynamic context subscription management

A. Shawky *, R. Olsen, J. Pedersen, H. Schwefel

Networks and Security, Department of Electronic Systems, Aalborg University, Denmark

ARTICLE INFO

Article history:
Received 13 January 2013
Received in revised form 31 July 2013
Accepted 22 October 2013
Available online 28 October 2013

Keywords:
Context
Awareness
Network
QoS
Optimization

ABSTRACT

Context-awareness is a key requirement in many of today's networks, services and applications. Context management systems are used to provide access to distributed, dynamic context information. The reliability of remotely accessed dynamic context information is impacted by network delay, packet drop probability, its information dynamics and the access strategy used. Due to the characteristics of the different access strategies, different levels of reliability of context information can be ensured, but at the same time, these strategies lead to different access traffic which impacts also the network performance, and hence feeds back to the reliability of the information. Furthermore, different levels of QoS may be available and used in order to mitigate the impact of network performance degradation on the reliability of the dynamic context information. In this paper we describe a system and algorithms that are capable of configuring effectively context access strategies in order to maximize reliability of all accessed dynamic context information. The framework utilizes and extends existing information reliability models, and it can utilize different network performance models. Simulation results of scenarios in which the framework uses finite-buffer bottleneck performance models demonstrate the effectiveness of our algorithm to increase reliability. Furthermore, the framework is applied to a scenario with QoS classes that allows to trade off delay and loss via different buffer-size configurations.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

The ability of applications to adapt to the users' environment is often referred to as context awareness, [1], and is becoming a key factor in today's mobile networks, since users need to be able to efficiently interact with applications and platforms in a highly dynamic world. Context awareness is achieved by accessing dynamic context information provided by context agents in a context management system and is a highly desirable feature for future mobile applications. However, the access to dynamic context information distributed in the environment has to be carefully designed to achieve scalability and context reliability. European projects like MAGNET Beyond [10], SPICE, [11] or E-SENSE, [12], and others outside Europe, have been researching and developing concepts for context management for some time, whereas reliability and accuracy indicators for context information just recently caught the attention e.g. in the project SENSEI, [13].

In those projects as well as within the research field, see e.g. [3,4], reliability of context information has been acknowledged as an important meta data to context information as the end user's satisfaction obtained from context-aware systems is directly depending on the reliability of the context information. In the respect of reliability, some work focuses on information reliability in terms of the uncertainty of the information source, e.g. using fuzzy logic approaches, [5] or [6], in which reliability is related to inexactness and uncertainty of obtained information and not the timely aspects of the information. Other work focuses on reliability of the information either by considering the age of the information, see e.g. [8,14] or [7]. However, the information age needs to be related to the

* Corresponding author. Tel.: +45 27821040.
E-mail addresses: asms@es.aau.dk, shawky84@gmail.com (A. Shawky).

1389-1286/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.comnet.2013.10.006

temporal dynamics of the information in order to provide a useful notion of reliability.

The reliability of context information is challenged by network delay, information dynamics, the context access strategy used and its parameter settings. The choice of the context access strategy on the other hand also influences network performance through the context access traffic. This paper provides a framework for automatic configuration of access to remote dynamic context information which accounts for the interplay between the network, the access strategies and their configurations, while maximizing the overall reliability of the information access.

The quality of context information plays an important role in adapting a system to continuously changing situations. In order for these systems to function optimally, they are required to implement measures that can resolve context conflicts in order to limit making of faulty context-aware adaptation decisions. Today there are several means to measure context quality and there is a large amount of literature on context modeling and representation, yet more needs to be done to insure context information exchange with guaranteed quality levels [23] [24]. There are several methods and techniques that are able to measure Quality of Context, amongst them are sensor training i.e (precision and correctness calculation), algorithms to calculate QoC value of aggregated context and conflict resolution algorithms. However most of the research done on context quality aims to develop a control scheme where context information is gathered, then classified and then is checked for quality by using one of the previous methods. Most of these methods seem to neglect the fact that there are networking issues that need to be addressed such as delay, overhead, jitter which directly context quality for dynamic context sources. While there is work [25] that provides networking QoS to context information, targeting higher-level context quality metrics with a model-based approach is the main novel contribution of this paper.

2. Context management framework

For applications to be able to adapt to their environment, access to dynamic context information [1], which describes the current situation, is required. Context Management systems offer flexible access often via dedicated query languages, e.g. [9,2], which allow applications easy access to distributed information.

A generic context management framework is illustrated in Fig. 1, which contains a set of entities called Context Agents (CA). Each CA is responsible for collecting data from its own local environment, as shown to the right in the figure: one CA exists on each device collecting information on, for instance, location, noise and temperature, respectively. A server in the network will act as context manager, which is responsible for maintaining the overlay network of context agents, i.e. it performs agent discovery, maintenance of agent information, information (de)registration and later on also access optimization for access to context information. In our considered architecture, context sensitive applications access context information through the context manager. The node that operates the context manager, is called the Context Management Node (CMN).

As seen to the right of Fig. 1, context information at the source changes according to external events, while the context-sensitive application would like to perform actions matching the true state of the context values at the source. When that information is dynamically changing, communication and processing delays can lead to deviations of the known context value at the application from the true value at the source. In our earlier work, we introduced the notion of mismatch probability (mmPr), [17] that is defined as *the probability that at the time instant of using a certain information for processing in the context-sensitive application, this information does not match the value at the (physical) source*. The reliability of context information has previously been acknowledged and considered an important part of the quality of context, [3,4,14] or [8], but not used in any effective way. In this paper we use this quantitative context reliability metric for intelligent choices of access strategies to improve reliability for all context sensitive applications requiring access to various dynamic distributed information elements.

The mismatch probability depends not only on the two stochastic processes (a) network delay and (b) information change process, but also on the strategy by which the information is accessed. The following three strategies will be used as principle mechanisms to access remote dynamic context information:

- *Reactive strategy*: whenever the application intends to process a certain context value, the context manager sends a request to the context providing agent, and gets a response with the information value in return. Fig. 2 illustrates this access strategy.
- *Proactive, event driven*: the context manager has setup a subscription to the context providing agent, and each time information changes value, the context agent sends an update to the context manager. For continuous information types, such change events can be defined via discretization intervals. The proactive event-driven strategy is illustrated in Fig. 3.
- *Proactive, periodic update*: the context manager has setup a subscription to the context providing agent, which after recurring time interval sends the current value to the context manager. Fig. 4 illustrates this periodic strategy.

The goal of the developed framework and algorithms is that the context manager is able to select and configure the access strategy by which it interacts with the various context agents to provide the information in such way that the reliability of the information is effectively increased.

The decision on which strategy to take is not trivial as it involves several stochastic processes and parameters that may be adjusted, and as each different strategy puts different loads to the network, and any decision effectively feeds back via an increased network delay to the mismatch probability and hereby may render decisions not optimal.

The objective of the algorithm is therefore to decide upon one of the three access mechanisms that should be used, as well as parameter settings, while considering that the context access traffic also affects the network; so impact on network delay and loss caused by the higher load is taken into account.

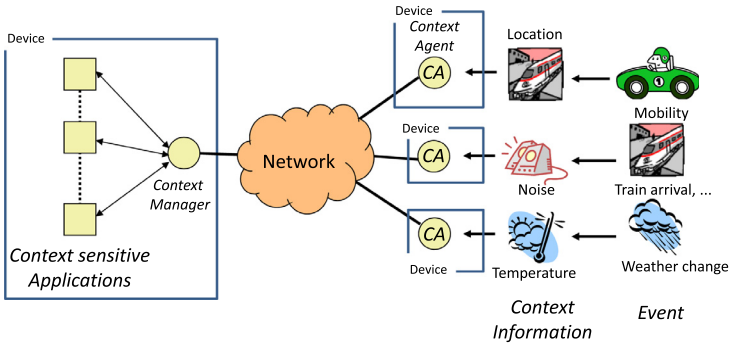

Fig. 1. Example of context management system – the context manager resides on the context management node.

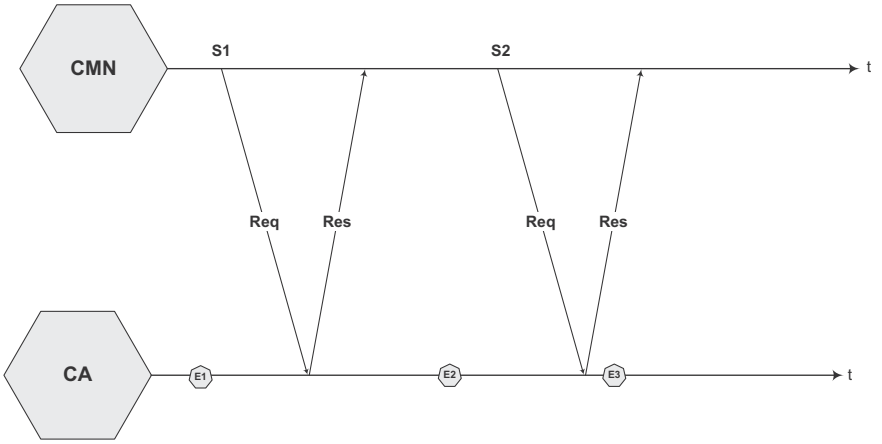

Fig. 2. Reactive strategy – the CMN sends request to the CA. The first request in the shown example leads to a matching information element, whereas the second ones leads to a mismatching information element. The delay for sending the response impacts the mismatch probability.

2.1. Interactions in the context management framework

The interactions between the different entities in the assumed context management framework are illustrated in Fig. 5. The different entities in the system are as follows:

- *Context agents*: Collect local context information, and provide remote access to context information. Context agents register at the CMN at start-up.
- *Context management node*: maintains a repository of (a) available context agents and (b) available context information in the network, and has possibility to control the interactions for context access between the context agents.
- *Network performance prediction*: a component that allows to obtain information of the network performance and provide information on how potential additional traffic changes network performance.
- *Context-dependent application*: This application uses dynamic context information for real-time decisions. The correctness of these decisions affects the resulting application quality.

Fig. 5 shows the basic message flow that occurs during the registration and subscription phases. The basic principles are:

- All CAs register at the CMN. They thereby provide information about the type of context elements they can offer and the relevant parameters. Particularly, they provide information on the temporal dynamics of the context element, which in our simplified setting are exponentially distributed inter-event times with a rate λ_i of the context-element i. Furthermore, also information on the size of update messages of the context element is provided.
- When an application or middleware function on a node is interested in a certain context element, it sends a subscription to the CMN. Along with other query related specifications, the application also sends its expected request rate and potentially also bounds on mismatch probability of the context element or on access delays.
- The CMN then evaluates which context access procedure with what parameter setting is best for this new subscription. In order to do so, it will interact with the network performance prediction function in the network. Having determined the most suitable configuration, the CMN informs the subscriber and the CA about this configuration and starts receiving context updates (in the proactive strategies) respectively relaying requests for context elements in the reactive strategies. All interaction between subscriber and CA is performed via the CMN in order to allow abstraction and efficient processing.

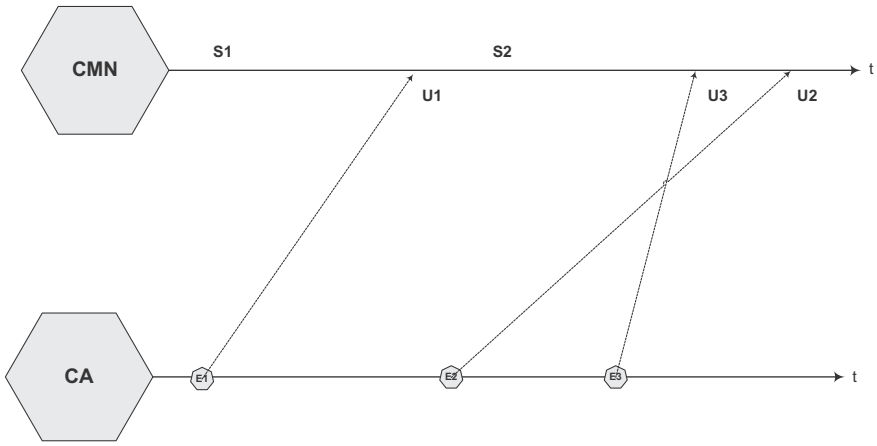

Fig. 3. Proactive event driven strategy – in the shown example, reception of the update from Event 1 leads to a correct value at the CMN until Event 2 happens at the CA. Update 2 will be filtered out at the CMN by the use of sequence numbers.

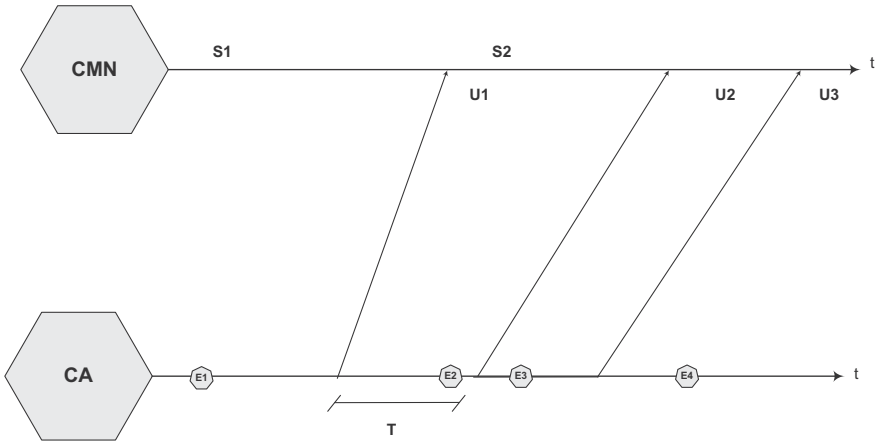

Fig. 4. Proactive periodic update – with a time interval T, an update of the most current value at the context agent is sent to the context manager. Reordered outdated updates are filtered out via sequence numbers.

- The CMN keeps track of all active subscriptions. Hence, as a variant, when a new subscription comes in, the CMN may not only determine an optimal access configuration for this newly arriving subscription, but it may in addition identify that it is beneficial to change some of the configurations of the already active subscriptions. Both cases will be specified and evaluated further below.

3. Analytic model representation

In order to calculate the impact of different access strategies on mismatch probability, different model parts need to be combined as illustrated in Fig. 6: Based on the parameters of the context elements accessed (change event rate λ and update size U_i) in the upper left and based on the request rate, μ, provided within the subscription request, the additionally created network traffic can be calculated in the upper left block. These are straightforward calculations, outlined further below for the individual strategies. The additional traffic is input to the network model, which is used to calculate network performance parameters for the paths between subscriber and context provider. In the example illustrated these parameters are mean delays and packet loss probabilities, but more advanced network models (beyond mean value calculations) are possible.

The delay and loss values are then used together with the context and subscription parameters to calculate the resulting mismatch probability per context element for different access configurations (lower left block). Finally, the calculated individual mismatch values are combined in a single metric called GmmPr as introduced further below and the model part outputs the minimum GmmPr and the corresponding strategy.

3.1. Utilization increase calculations

When a new subscription is executed by the CMN, additional network traffic is generated. The required additional throughput, η, is calculated in the following for the three different access strategies. U is thereby the size of the different messages (requests or updates), μ is the request

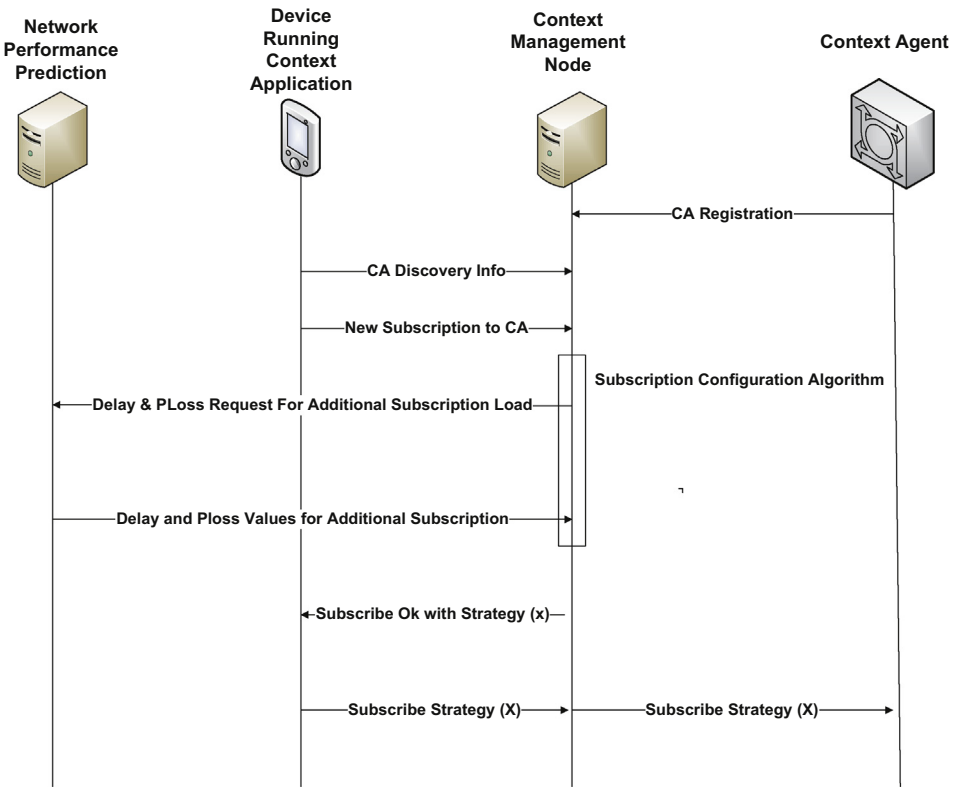

Fig. 5. Interactions with the CMN during establishment of a context subscription.

rate, λ is the rate of the change events of the context information, and τ is the rate of the periodic updates. Messages that are dropped in the network are included in the throughput calculations.

Reactive strategy. The context access traffic generated by the reactive strategy is depending on the request size, the response size and the rate of requests (assuming request arrivals with rate μ). The generated traffic is:

$$\eta_{rea} = \left(\mathbb{E}(U_{rqs}) + \mathbb{E}(U_{rsp})\right) \cdot \mu.$$

Proactive event driven update. For the proactive event driven strategy the traffic generated depends entirely on the change rate of the information and the size of the updates:

$$\eta_{evn} = \mathbb{E}(U_{update}) \cdot \lambda.$$

Proactive periodic update. For proactive periodic update strategy the context access traffic generated is directly depending on the update rate and the size of the updates:

$$\eta_{per} = \mathbb{E}(U_{update}) \cdot \tau$$

3.2. Network model

The Network model is used to translate utilization into delay and packet loss probability. This paper uses two different network models: (1) an M/M/1/K model, which has the advantage of a small parameter space, while still allowing to analyze trade-offs between delay and loss when varying the buffer-size K. (2) A more complex network model consisting of the convolution of a transmission delay with a queueing delay obtained from a bursty MMPP/M/1/K model.

The mean delay and packet drop probabilities at utilization ρ of the M/M/1/K queue are found in any standard queuing theory book:

$$Packet\ Loss = \frac{((1-\rho)\rho^K)}{(1-\rho^{K+1})}$$

$$Queue\ Length = \frac{\rho}{(1-\rho)} - \frac{(K+1)(\rho^{K+1})}{(1-\rho^{K+1})}$$

$$Delay = \frac{Queue\ Length}{\Lambda(1-P_{loss})}$$

Λ is hereby the overall offered load to the queue (in packets per time unit), K the size of the buffer at the bottleneck (measured in packets).

The analysis of the algorithms for context subscription handling also uses a more complex network model which is formed from the convolution of an end-to-end forwarding delay and a queueing delay. The forwarding delay is modelled by an exponential delay. The queueing delay is assuming bursty cross-traffic with exponential ON–OFF pattern superimposed to Poisson context access traffic. The cross-traffic represents a base-load, while the Poisson rate of the context traffic is a function of the number of active subscriptions, their context parameters and the selected access strategies, see Section 3.1. This traffic model

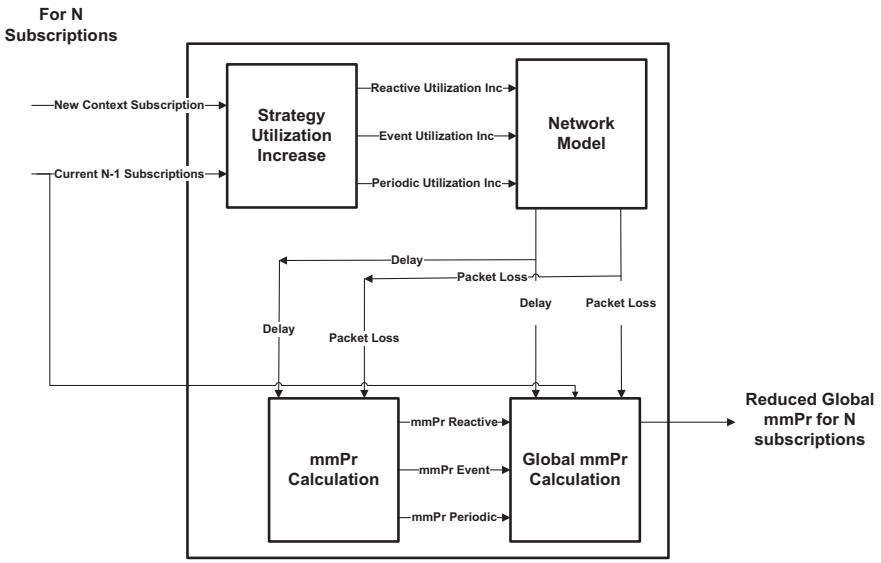

Fig. 6. Model overview of internal parts of the algorithm that is being executed at the CMN.

can be represented by a 2-state Markov-modulated Poisson Process (MMPP) and the performance metrics, mean delay and packet loss probability, of the corresponding MMPP/M/1/K queue can be calculated with matrix-analytic methods [21]. In order to keep the complexity of the mmPr calculations low, the queueing delay is then assumed to be exponentially distributed with a mean that is obtained from the numerical solution of the MMPP/M/1/K queue. That way, the mismatch probability calculations can be performed based on Erlangian-2 type distributions for network delay. Such distribution is a specific case of a general phase-type or matrix-exponential distribution, represented in the notation introduced by Lipsky [22] by a vector/matrix pair $\langle \mathrm{p}, \mathrm{B} \rangle$, such that its cumulative distribution function and density function can be described as

$$F(t) = 1 - \mathrm{p}\exp(-t\mathrm{B})\varepsilon', \quad f(t) = \mathrm{pB}\exp(-t\mathrm{B})\varepsilon'.$$

Here p is an entry row-vector to the delay model, B is the generator matrix for the delay model and ε' is a column vector of ones. In the simplest case of an exponentially distribution, $\mathrm{p} = 1$ and $\mathrm{B} = v$ (the rate of the process), which later simplifies the used mismatch probability equations. The used Erlangian-type model for the network delay has a Matrix exponential representation:

$$\mathrm{p} = [1\ 0], \quad \mathrm{B} = \begin{bmatrix} v_t & -v_t \\ 0 & v_q \end{bmatrix} \tag{1}$$

where v_t is the transmission delay rate, and v_q the queuing delay rate calculated from the queueing model as described above.With this, it is fairly simple to independently adjust the two types of delay occurring in the system we investigate. More complicated delay models can easily be represented by more general matrix-exponential distributions.

3.3. Mismatch probability calculations

We extend the equations of [17] that describe the mismatch probabilities for the different access strategies. The main extensions are to include packet loss, which are relevant when UDP transmissions are used for the context access. In case of context access traffic via TCP, packet losses lead to retransmissions, which can me mimicked by an adjusted message delay distribution. We however assume UDP context traffic for which losses are relevant to be included in the model.

Reactive strategy. In order to receive a correct context information message, the reactive strategy requires successful transmission of two messages; otherwise the CMN does not receive a response which we assume to lead to a mismatch. Under the condition of successful two way communication, i.e. the request and response are not lost, it is only the delay of the response that may lead to mismatches. Extending the approach of [17] while assuming both the inter-event time, $\langle \mathrm{p}_E, \mathrm{B}_E \rangle$, and the network delays, $\langle \mathrm{p}_D, \mathrm{B}_D \rangle$, are Matrix-exponential distributed renewal processes, we can express the mismatch probability as follows:

$$mmPr_{rea,ME} = 1 - (1 - p_{loss})^2 \times \frac{(\mathrm{p}_E\mathrm{B}_E^{-1}) \otimes (\mathrm{p}_D\mathrm{B}_D)}{\mathrm{p}_E\mathrm{B}_E^{-1}\varepsilon'_E} [\mathrm{B}_E \oplus \mathrm{B}_D]^{-1}\varepsilon'_{dim(B_E)\cdot dim(B_D)} \tag{2}$$

The symbols $\otimes$ and $\oplus$ represent the Kronecker-product and Kronecker-sum, see [20]

When both the network delay and the time between context change events are exponentially distributed with rates v and λ, this equation simplifies to:

$$mmPr_{rea,exp} = 1 - (1 - p_{loss})^2 \frac{v}{\lambda + v}.$$

Note that the mmPr increases with increasing context change rate λ and with increasing network delays (which

correspond to decreasing rate, v, of the delay distribution). For recurrent context information types, the equations from [17] can be analogously extended. In addition to the reactive strategy, it would also be possible to cache data that has been received to be used as response for subsequent request within some finite time interval. This mechanism we evaluated in [18], however, to keep the configuration space low, we choose not to allow caching techniques to be applied.

Proactive event driven update. The mismatch probability without packet loss for this approach turns out to be exactly the same as for the reactive strategy, see [17]; in case of potential packet loss, a information match now requires that the last update message has not been lost. We assume here that information updates are complete, i.e. each update completely replaces old information stored at the context manager. An alternative update approach does exist, where updates are incremental, but with a non-zero packet loss probability, this type of update is not useful since one dropped message leads to subsequent mismatches. For the case of Matrix-exponential inter-event times and network delays, the results from [17] are hence extended to

$$mmPr_{event,ME} = 1 - (1 - p_{loss}) \times \frac{(\mathrm{p}_E \mathrm{B}_E^{-1}) \otimes (\mathrm{p}_D \mathrm{B}_D)}{\mathrm{p}_E \mathrm{B}_E^{-1} \varepsilon'_E} [\mathrm{B}_E \oplus \mathrm{B}_D]^{-1} \varepsilon'_{dim(B_E)\cdot dim(B_D)}, \quad (3)$$

which reduces in case of exponential distributions to

$$mmPr_{event,exp} = 1 - (1 - p_{loss}) \frac{v}{\lambda + v}.$$

Proactive periodic update. The mmPr model of the periodic update case assumes an exponentially distributed update interval with rate τ in order to simplify the mathematical calculations. If context providing processes are lowly prioritized by the operating system, stochastic fluctuations in the update period are not rare, even if timers are used. The basic approach and the resulting equation to calculate the mismatch probability is the same as in [17], and for Matrix Exponential distribution can be expressed as

$$mmPr_{per,ME} = \frac{e^{\tau \overline{D}}}{\overline{E}} \int_0^\infty e^{-\tau t} exp\left[-\tau \mathrm{p}_D \mathrm{B}_D^{-1} exp(-\mathrm{B}_D t) \varepsilon'_D\right] \cdot \left[\mathrm{p}_E exp(-\mathrm{B}_E t) \varepsilon'_E\right] dt, \quad (4)$$

which can be simplified in case of exponentially distributed processes

$$mmPr_{per,exp} = \phi e^{\psi} \frac{\Gamma(\phi + \psi)}{\psi^{\phi+\psi}} F_{\Gamma(\phi+\psi,\psi)}(1).$$

The latter equation has the advantage that it is integral free, but is limited to only the exponentially distributed case. For the two stage delay, numerical integration is required for solving the mmPr.

Including packet loss is done by considering the update process as a thinned Poisson process with thinning probability $(1 - p_{loss})$, so that the parameter ψ now becomes $\psi = \tau(1 - p_{loss})/v$ (where τ is the update rate, and v is the delay rate). The other parameter remains as $\phi = \lambda/v$ (ratio of event and delay rates). $F_{\Gamma(a,b)}$ is the cdf of a gamma distribution with parameters a and b.

Rejection of context requests. As context subscription traffic in any of the above three configurations also influences network delays and loss probabilities, it will increase the mmPr for other subscriptions. Depending on the optimization target, it can hence be beneficial to reject context subscriptions to avoid this negative impact on other ongoing subscriptions. In case of rejection of a subscription, no additional network traffic is generated, but the mismatch probability for the context accesses of this rejected subscriptions are $mmPr = 1$.

3.4. Global mismatch probability

When there are N context subscriptions, we define in a single metric that we call the global mmPr (GmmPr). For simplicity in this work we use the average mmPr of the N sources:

$$GmmPr := \sum_{i=1}^{N} \frac{mmPr_i}{N}.$$

Other definitions, e.g. using weighted averages based on context access frequency or based on context relevance can be considered when demanded by the specific scenario.

4. Optimization of dynamic context subscriptions

The modelling framework of the previous section is now utilized for configuration selection of context subscriptions within the context management framework introduced in Section 2.

4.1. Algorithm descriptions

In the following, we present two algorithms that run at the CMN. The first algorithm provides the base case, when a subscription is initiated by a context-dependent application or device middleware, the CMN determines the optimal configuration for this newly starting subscription while taking into account the network conditions and also the impact of this new subscription traffic on the already ongoing context subscriptions. The second algorithm includes a reconfiguration of already active context subscriptions and as such is an extension of the first one. Although we do not consider this later in the evaluation, the second algorithm could also be triggered by changes in the network resources, e.g. by increase or decrease of the other traffic in the network, or by reduction of available bandwidth/change of delays in wireless communication settings.

Algorithm 1: Configuration of new subscriptions. The first algorithm is executed by the CMN upon the reception of a new subscription. As our intention here is to evaluate the benefit from an intelligent configuration choice, the algorithm is not optimized for efficiency, but rather implements a brute-force search over the possible configuration set – taking parameterized strategies (in our example the

periodic strategy with update rate τ) via discretized search spaces into account.

Algorithm 1. New subscription configuration algorithm

```
function determine_configuration_of_new_subscription
Input: Current network load L and available network (bottleneck) resources R
               in the path context agent - CMN - subscriber.
        Request Rate mu_i of new subscription.
    Available Information (persistently maintained at CMN)
           From registration: Context-value change rate lambda_i, and update size u_i of
           context element of new subscription.
           From previous subscriptions: Parameters and configurations of
                     currently active subscriptions.

Output: access strategy for new subscription
           parameterizations of the access strategy
           (here: rate of proactive periodic strategy).
           resulting new GmmPr best_mmPr.
Begin
    best_mmPr = 1.0;
best_strategy = not_defined;
For all access strategies in (Reject, reactive, proactive event-driven,
                     proactive periodic_ratel, ...proactive periodic_rateK);
       calculate additional network load L_inc created by NEW subscription
                     when using this access strategy (see Section III);
       Call network model to calculate network performance metrics (D, p_l)
              at load L + L_inc for resources R;
       Calculate GmmPr for these network performance metrics and
                        existing plus new subscriptions;
       If calculated GmmPr < best_mmPr;
              Set currently considered access strategy and parameters
                                 as best one;
              Update best_mmPr;
    end-if
end for-loop;

end function;
```

Algorithm 2: Reconfigure all subscriptions. While the previous algorithm only considers the newly incoming subscription, it may also be worth to adapt the way already existing subscriptions are executed. As the CMN keeps track of all subscriptions, their access strategies and their parameters, it can use this information to optimize the target metric further. As we are only interested in a potential benefit of such reconfiguration, we also in this case do not implement very sophisticated efficient search approaches, but rather an algorithm that first determines the optimal configuration of the new subscription, and then iterates over all existing subscriptions to check if an improvement can be obtained by changing the configuration. This obviously does not guarantee convergence to the true optimum, as it locally optimizes each individual subscription. Nevertheless, the algorithm is useful in the evaluation part to investigate benefits from reconfigurations.

Algorithm 2. Subscription reconfiguration algorithm

```
function iterate_over_existing_Subscription
 Input: Current network load L and available network (bottleneck)
          Number of repetitions, rep, for iterative search.

Available Information (persistently maintained at CMN)
          From previous subscriptions: Parameters and configurations of
                    currently active subscriptions.
Output: optimized access strategies for existing subscriptions
          parameterizations of the access strategies
          (here: rate of proactive periodic strategy).
          resulting new GmmPr best_mmPr.

Begin

best_mmPr = 1.0;
best_strategy = not_defined;

Repeat rep times,

       for all previously existing subscriptions
                    subtract current load L_old caused by this subscription in
                          current access mode from total load L.
                    Take this subscription out of the currently active set and
                          consider it as 'new';
                    determine_configuration_of_new_subscription at load L-L_old
                          considering the just taken out old subscription;
                    Add subscription to the currently active set with potentially changed
                              access strategy;
                    L = L-L_old+(load created by reconfigured subscription)
                    Call network model to calculate network performance metrics (D, p_l)
                                  at load L + L_inc for resources R;
                              Calculate GmmPr for these network performance metrics and
                                    existing plus new subscriptions;
                              If calculated GmmPr < best_mmPr;
                                  Set currently considered access strategy and parameters as best
  one;
                                  Update best_mmPr;
                              end-if
             end for-previously-existing;
end repeat-rep-times;
```

Note that the context access strategies are actually not updated at every individual context request, but rather at establishment of new subscriptions. Once a subscription has been accepted, the CMN and CA communicate in the determined way for a number of context requests (or change events or periodic update messages). Only when a new subscription is signaled to the CMN, a reconfiguration may happen. The network load created by the reconfigurations themselves is therefore small in case of long lasting subscriptions; the network models in this paper therefore do not consider the subscription management and subscription reconfiguration traffic.

4.2. Evaluation methodology

To evaluate the algorithms' performance a series of simulations were carried out. These simulations will compare the intelligent mmPr-based configuration optimization algorithms against strategies that assign a constant default access strategy to new subscriptions. The simulation considers the arrival of N context access requests at the CMN, the request parameters as well as the context parameters are thereby stochastically varying (using uniform distributions). All parameters and their values in the simulation are listed in Table 1. The simulation evaluation

Table 1
Context information parameters.

Context parameter	Value
Context request rate	[0.2–1] Requests/Sec
Context event rate	[0.2–1] Events/Sec
Context update rate range	[1–10] Updates/s
Context request size	200 Bytes
Context update/response size	[800–1200] Bytes/Update
Packet size	1000 Bytes

Table 2
Network model parameters, M/M/1/K.

Service rate (μ)	10 MBit/s (1250 pck/s)
Buffer size (K)	100 Packets
Cross traffic	9 MBit/s

Table 3
Network model parameters, MMPP/M/1/K.

Service rate (μ)	10 MBit/s (1250 pck/s)
Buffer Size (K)	100 Packets
Cross traffic rate during OFF	0 MBit/s
Cross traffic rate during ON	10 MBit/s (1250 pck/s)
Mean ON time	0.15 s
Mean OFF time	0.05 s
Exponential Transmission Delay, mean	500 ms

assumes that the parameters of the specifically addressed context elements (change rate, update size) and the resulting individual request rate are known by the context agent respectively the subscribing application and hence are correctly specified (see Tables 2 and 3).

For the network, we assume that the context access traffic of all N subscriptions share the same network bottleneck, we consider a scenario where a CMN is running on a wireless connected device, hence the CMN receives and manages these N subscriptions over the same wireless access interface. We use two different models: (1) a network model based on solely the queueing delay of an M/M/1/K model and (2) a network model which uses an exponential forwarding delay in convolution with a queueing delay whose mean is calculated from a bursty MMPP/M/1/K model. The service rate used for both models is motivated by a WLAN access scenario with a link capacity of 10 Mb/s. The buffer-size for the used bottleneck link is set to $K = 100$ messages. It is furthermore assumed that the node is not exclusively running the CMN functionality, but other (non-context) traffic is utilizing the same wireless interface, creating a 'base load' taking two different values, bursty traffic with 75% utilization for the MMPP/M/1/K model, and 90% for the M/M/1/K model.

The mmPr models are then used in the simulations to calculate the resulting GmmPr of the context access after each incoming subscription, comparing the selections from the algorithms from the previous section with three default constant access configurations. The comparison cases are: (1) always reactive access; (2) always event-driven access; and (3) periodic with a fixed update rate of $\tau = 0.2$ updates/s. Obviously, other heuristic methods of selecting access could be considered, but are not included here due to space limitations.

The algorithm that minimizes the GmmPr based only on the configuration of the newly incoming subscription (Alg 1) is compared with the approach that iterates over all active subscriptions (Alg 2), where the number of iterations is set to 2. Table 1 summarizes the context information parameters and the M/M/1/K and MMPP/M/1/K parameters used for all six scenarios. The context parameters are thereby uniformly distributed in the range specified by the given intervals in the table.

4.3. Simulation results

M/M/1/K network model. Fig. 7 shows the mean GmmPr of 20 simulations each having 50 subscription requests. The corresponding dashed lines show the 95% confidence intervals of this mean estimate from the 20 simulations. The graph represents a comparison of the fixed access strategies, periodic, reactive, and event-driven, with Algorithm 1 (optimized configuration of newly incoming subscription only) and the iterative Algorithm 2, which optimizes the configuration of the newly incoming

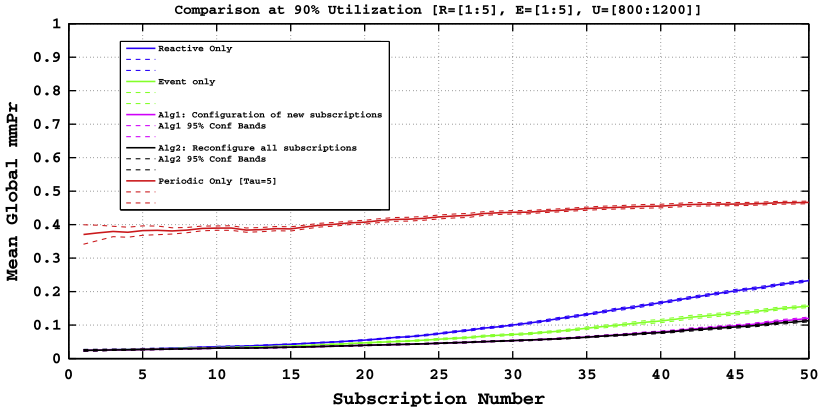

Fig. 7. Average global context mismatch probabilities when using an M/M/1/K model with 90% cross-traffic utilization, using an update rate of 1/5 updates per second as comparison.

subscription and subsequently makes two iterations over the already established subscriptions to reconfigure in case of better configuration options. Rejections of access requests (leading to mismatch equals 1 for this subscription) do not occur. After 50 subscriptions at the right end of the curve, averaged over the 20 runs of Algorithm 1, the configurations that result are: 36.4 event driven, 13.6 reactive, 0 periodic. The reconfiguration of already previously accepted subscriptions (Algorithm 2, black curve) shows a slightly lower average GmmPr compared to only selecting a configuration for the new subscription (Algorithm 1, pink curve). Algorithm 2 configuration results are: 31.45 event driven, 18.55 reactive and 0 periodic. The results show that the model-based optimizations lead to a substantially lower GmmPr than the fixed strategies. The iteration with potential reconfigurations of already existing subscriptions thereby shows a slight further gain, which is statistically relevant judging from the detailed inspection of the confidence intervals at slightly lower confidence level; however, the additional computational effort and also communication effort for reconfiguration of subscriptions appears not justified in the considered scenario.

Network model based on convolution of transmission time with queueing delay from bursty model. Fig. 8 shows the mean GmmPr of 20 simulations each having 50 subscription requests when using the more complex network model, consisting of an exponentially distributed transmission delay convoluted by a queueing delay whose mean is determined from a MMPP/M/1/k model with increasing utilization upon accepted subscriptions. The service rate is 10 Mb/s and the Cross traffic is 7.5 MB/s. The corresponding dashed lines show the 95% confidence intervals of this mean estimate from the 20 simulations. The graph again shows a comparison of the fixed access strategies with Algorithm 1 (optimization of newly incoming subscription) and the iterative Algorithm 2 with two optimization iterations (including potential reconfigurations of previously accepted subscriptions). Rejections of access requests do not occur. After 50 subscriptions at the right end of the curve, averaged over the 20 runs of Algorithm 1, the configurations that result are: 44.85 event driven, 5.15 reactive, 0 periodic. The reconfiguration of already previously accepted subscriptions (Algorithm 2, black curve) shows a slightly lower average GmmPr compared to only selecting a configuration for the new subscription (Algorithm 1, pink curve). Algorithm 2 configuration results are: 31.8 event driven, 18.2 reactive and 0 periodic.

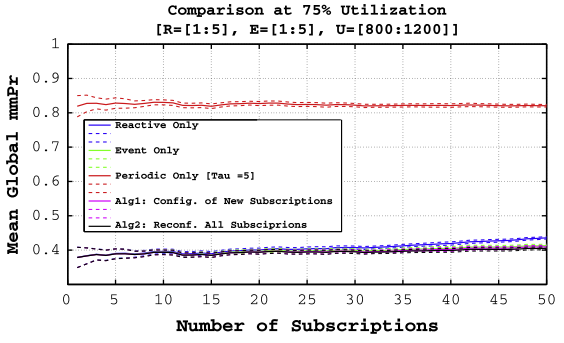

Fig. 8. Average global context mismatch probabilities when using the more complex network model with 75% utilization from bursty cross-traffic.

It is of interest to notice the big difference in the numerical values of GmmPr in the two cases using the different network models. Adding the exponential transmission delay (which is independent of network traffic) in the second scenario, the impact of increasing number of subscriptions becomes much lower. On the other hand, the absolute values of the GmmPr are much higher than for the M/M/1/K case, despite the substantially lower utilization. The optimization algorithm is still effective in the scenario with complex network model, but in the chosen case, a fixed event-driven strategy almost performs equally well. Note however, that this result is a consequence of the considered scenario, and the algorithm would still be needed to identify the choice of strategies in a general setting. The benefit from reconfigurations of already existing subscriptions is small in both scenarios. That algorithm however is still useful as it can be applied also in settings when network behavior changes drastically (e.g. due to path failure and re-routing), since then existing subscriptions may need to be adapted.

4.4. Summary

The model-based optimization algorithms are executed on the Context-Management-Node (CMN), which is the node that receives all context subscriptions and manages the interaction with the context providers. We introduced two algorithm versions: (1) An algorithm that makes decisions on what context access strategy and configuration to use when handling a newly incoming subscription and (2) an algorithm that in addition considers reconfiguration of already established context subscriptions. The target in both cases is a metric called Global mismatch Probability (GmmPr) that is the average over all subscriptions of the probability that at the time of processing context information, the physical context source has changed value. The model used in both algorithms addresses the feedback loop of the impact of decisions on network traffic and hence network performance. Simulations were made to test and compare the algorithms. The simulation results showed that selections of context access configurations based on the mmPr model are able to reduce GmmPr, however, the level of improvements depends highly on the scenario considered.

5. Subscription configuration for class-based traffic differentiation

Modern packet-switched network realizations allow for differentiated treatment of different traffic types, e.g. via the DiffServ architecture [16]. That way it is possible to apply different scheduling strategies and to configure different buffer-sizes for queues in bottleneck routers. The latter degree of freedom allows to trade-off delay for packet-loss. In order to extend the context subscription configuration algorithms from the previous section to a

traffic differentiation scenario, we first motivate the expected benefit from class-based traffic treatment by an analysis of the trade-off of loss versus delay on the context mismatch probability. We subsequently extend Algorithms 1 and 2 to the class-based scenario and investigate the resulting context subscription performance, measured by global mismatch probability, in simulation experiments.

5.1. Delay versus loss trade-off

In a class-based QoS differentiation scheme, it is possible to have different subscriptions treated in different classes which potentially use different buffer-sizes. Using short buffers, lower transmission delays result at the cost of increased packet loss. Using the mmPr models and assuming the M/M/1/K bottleneck buffer model, the impact of the buffer size changes on the resulting context quality can be analyzed in the analytic model. Fig. 9 shows the mmPr of the reactive access strategy, when it operates in a network which is described by a bottleneck M/M/1/K queuing model with varying buffer size K. The different curves show from bottom to top different context change rates of $\lambda = 1, 2, 5$, while other parameters of the context information (update size equals to one single packet) and the context access rate ($\mu = 2$) are constant, and the service rate of the M/M/1/K model maps a 10 Mb/s WLAN scenario with packet size of 1000 bytes, which here is assumed to run in overload at $\rho = 1.2$. Note that in order to analyze the delay-loss trade-off, in contrast to the previous algorithms and evaluations, the additional (constant) context access traffic is here not taken into consideration, so the bottleneck utilization is exactly the same across all strategies.

Fig. 9 shows that for the chosen example scenario, each curve reaches a minimum mmPr value, marked by the stars on the curves. As we are using analytic formulas, the buffer-size K_{min} at which the minimum mmPr is achieved can be calculated from numerically searching the root of the mmPr derivative for the given parameters. Fig. 10 shows the resulting K_{min} values for different event rates (x-axis) and for different utilization of the bottleneck queue. For the reactive strategy with the used scenario parameters, scenarios with lower event rates and lower utilization favor delay over loss, so larger buffers are better (K_{min} grows). Note that this section is not intending to formally proof such behavior but rather acts as motivation that DiffServ-like class differentiation with different buffer sizes can be valuable and that there exist approaches to determine optimal class parameters.

5.2. Algorithm

Basic approach for a class-based scenario is to extend the algorithms of the previous section by also iterating over the possible C DiffServ classes. The Network model then needs to be able to compute the resulting performance (mean delay and loss probability in our case) given that a certain context traffic is added to a specific class. To keep this part simple in our analysis, we assume a static bandwidth assignment, i.e. the overall link capacity of the bottleneck is distributed to the C classes, so that for each class i, an M/M/1/K_i model can be applied to calculate mean delay and loss within that class. The buffer-size K_i and the service rate μ_i can then be set independently for each class, as long as $\sum_i \mu_i = \mu$.

As before in Section 4, the extended algorithm is executed at the reception of a new subscription. The algorithm then checks the available QoS classes and calculates the mmPr over all access strategies, now also for all QoS classes. The algorithm then selects the class and the strategy that results in the lowest GmmPr, taking into account also the impact of the additional context traffic on the already existing subscriptions (only the subscriptions in the same class as the new subscription are affected in the used scheduling strategy). Analogously to Algorithm 2, subsequently already existing subscriptions are reconsidered for potential reconfiguration. The extended pseudo code is found below.

Algorithm 3. Diffserv Subscription Classification algorithm

```
function determine_QoS_class_and_access_configuration_of_new_subscription
Input: Current load L_j on classes and available network (bottleneck) resources R_j for each class j
            in the path context agent - CMN - subscriber.
       Request Rate mu_i of new subscription.

Available Information (persistently maintained at CMN)
          From registration: Change rate lambda_i, and update size u_i of
          context element of new subscription.
          From previous subscriptions: Parameters and configurations of
                currently active subscriptions.
Output: QoS Class for new access strategy
                    access strategy for new subscription
        parameterizations of the access strategy
        (here: rate of proactive periodic strategy)
```

```
        resulting new GmmPr best_mmPr.
Begin

best_mmPr = 1.0;
best_class = NaN;
best_strategy = NaN;
for all classes j = 1, ,N;
   For all access strategies in (Reject, reactive, proactive event-driven,
          proactive periodic_ratel, ...proactive periodic_rate);
      calculate additional class load Lj_inc created by NEW subscription
                when using this access strategy (see Table I);
      Call network model to calculate class performance metrics (Dj, Pj)
                  at class load Lj + Lj_inc for resources R;

      Find Minimum N Class GmmPr

      If calculated GmmPr < best_mmPr;
         Set currently considered access strategy and parameters
         as best one;
         best_Class = j;
         Update best_mmPr;
   end-if
  end for-loop-access strategies;
  Calculate Link Global mmPr (GmmPr) for all classes;
   Select class, strategy and parameter that minimizes Link GmmPr;
   If Subscription_number > 1 for selected class
   Run Reconfigure subscriptions and recalculate GmmPr;
   end for-loop-classes;
end function;
```

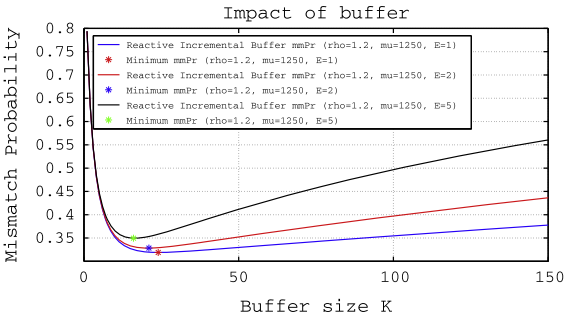

Fig. 9. Impact of buffer size in the M/M/1/K network model on mismatch probability of different access strategies at high utilization.

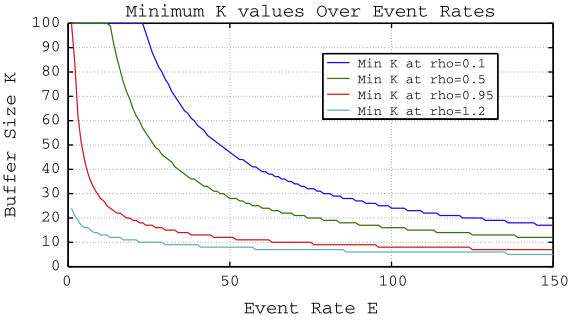

Fig. 10. Buffer-size K_min that minimizes the mmPr for different change rates of the context information (x-axis) and different utilization (different curves).

5.3. Simulation model

In order to evaluate the algorithms, simulations as described in Section 4.2 were extended to the class-based scenario enabling now to compare Algorithm 3 with the class-less scenario using Algorithm 2. As in Section 4, the simulation considers the arrival of N context access requests at the CMN. All parameters and their values in the simulation are listed in Table 4. Same assumptions as for the previous simulations hold: (1) The CMN is assumed to have perfect knowledge of the average request rates for the individual subscriptions. (2) For the network, we assume that the context access traffic of all N subscriptions share the same network bottleneck. (3) A WLAN scenario is used to motivate the network parameters. The corresponding same packet rate is in the class-based scheme mapped to the larger Class 1, leading to class utilizations from background traffic of $\rho_1 = 1$ (while Class 2 has no background load, $\rho_2 = 0$) for the 90% utilization scenario. For the 100% utilization scenario the resulting background utilization on Class 1 is $\rho_1 = 1.11$. Mapping all background traffic into Class 1 corresponds to a scenario in which Class 2 is exclusively reserved for context traffic, while the subscription configuration algorithm is also allowed to use Class 1.

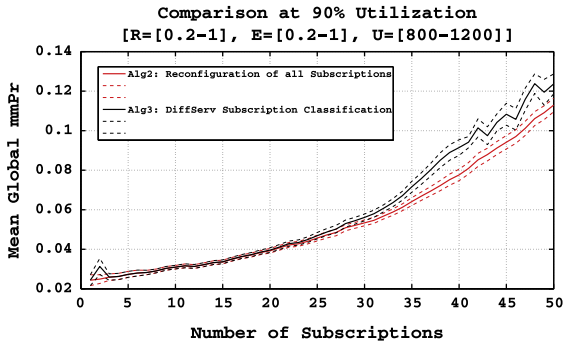

Fig. 11. Classfull vs classless comparison at 90% background traffic in the M/M/1/K network model.

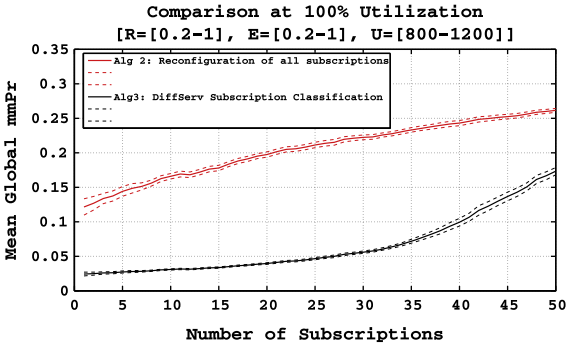

Fig. 12. Classfull vs classless comparison at 100% background traffic in the M/M/1/K network model.

Table 4
Context information parameters.

Context parameter	Value
Context request rate (μ)	[0.2–5] Requests/s
Context event rate (λ)	[0.2–5] Events/s
Context request size (U_{rqs})	200 Bytes
Context update rate (τ)	[1–20] updates/s
Context update/response size (U_{rsp})	[800–1200] Bytes/ Update
Classless buffer size (K)	100 Packest
Class 1 buffer size (K_1)	100 Packets
Calss 2 buffer size (K_2)	30 Packets
Classless maximum link capacity (μ_{link})	10 Mbit/s
Class 1 maximum link capacity (μ_1)	9 Mbit/s
Class 1 maximum link capacity (μ_2)	1 Mbit/s

5.4. Simulation results

Scenario with 90% utilization from other traffic. Fig. 11 shows the Comparison between classless context optimization and class-based context optimization with a base-load of 9 Mbit/s, which in the class-based scheme is put onto Class 1. The results shows that the class distribution for 20 simulation runs is: on average 3 out of 50 subscriptions are in Class 1, while Class 2 is used for on average 47 out of 50 subscriptions; no subscriptions are rejected. The subscriptions show on average the following distribution in the class-full scenario: 18.75 using the reactive strategy, on average 31.15 using the proactive event strategy, and 0.1 periodic.

For the classless scenario the same results as shown in Section 4.3 apply. As it can be seen from the results, splitting the link into two classes does not benefit the reduction of Global mmPr for active subscriptions in the shown scenario of 90% background traffic utilization.

Scenario with 100% utilization from other traffic. Fig. 12 shows the comparison between classless context optimization and class-based context optimization with a base-load of 10 Mbit/s which again is assigned to Class 1 in the class-based scheme. The results shows that for the class-based scenario, the class distribution is on average 0.2 subscriptions for Class 1 and 49.8 subscriptions for Class 2. As for the subscription distribution, for the class-based scenario, it is on average 0.7 rejects, 18.55 reactive, 30.55 proactive event driven and 0.2 for proactive periodic update. The results for the classless scenario were presented in Section 4.3. In this highly saturated scenario it can be seen that using a split link benefits the Global mmPr for all active subscriptions.

6. Summary

We investigated the possibility of increasing context information reliability by configuration of access strategies during subscription establishment and by use of QoS classification. The paper defines and evaluates an algorithm which is intended to be used as a part of a context QoS control framework. The algorithm uses extensions of analytic calculations of mismatch probabilities to scenarios with packet loss, motivated by UDP based context access scenarios. Furthermore, it uses a network model to compute the impact of additional context traffic on network performance metrics. The model-based algorithm is executed on the Context-Management-Node (CMN), which is the node that receives all context subscriptions and manages the interaction with the context providers. The algorithm was evaluated for two network models; a simple M/M/1/K bottleneck queuing model and a more complicated network model that uses a convolution of transmission delays and load-dependent queueing delays, in this case for bursty ON/OFF background traffic overlaid to Poisson context traffic. The evaluation results show the effectiveness of the approach. Evaluations of DiffServ like scenarios with two traffic classes allow to quantify the benefit of utilizing dedicated network resources for context traffic.

References

[1] A.K. Dey, Providing Architectural Support for Building Context-Aware Applications, Ph.D. thesis, Georgia Institute of Technology, 2000.
[2] Luis Sanchez, Rasmus L. Olsen, Martin Bauer, Marc Girod Genet: A Generic Context Management Framework for Personal Networking Environments, First International Workshop on Personalized Networks, 2006.
[3] T. Buchholz, A. Kupper, M. Schiffers, Quality of context: what it is and why we need it, in Proceedings of the 10th Workshop of the OpenView University Association: OVUA'03, Geneva, Switzerland, July 2003.
[4] C. Villalonga, D. Roggen, C. Lombriser, P. Zappi, G. Trster, Bringing quality of context into wearable human activity recognition systems, in: Proceedings of the First International Workshop on Quality of Context (QuaCon 2009), LNCS 5786, 2009, pp. 164–173.

[5] G. Stoilos, G. Stamou, V. Tzouvaras, J.Z. Pan, I. Horrocks, Fuzzy OWL: uncertainty and the semantic web, IEEE Intelligent Systems 21 (5) (2006) 84–87. ISSN:1541-1672.
[6] D. Preuveneers, Y. Berbers, Quality extensions and uncertainty handling for context ontologies, in: Proceeding of Context and Ontologies: Theory Practice and Applications, Riva del Garda, Italy, August 28, 2006.
[7] K. Henricksen, J. Indulska, A. Rakotonirainy, Generating context management infrastructure from high-level context models, in: Proceedings of 4th International Conference on Mobile Data Management, 2003.
[8] S. McKeever, J. Ye, L. Coyle, S. Dobson, A context quality model to support transparent reasoning with uncertain context, QuaCon 2009, LNCS 5786, Lecture notes published by Springer-Verlag Berlin Heidelberg, 2009, pp. 65-75.
[9] A. Devlic, E. Klintskog, Context retrieval and distribution in a mobile distributed environment, in: Proceedings of 3rd Workshop on Context Awareness for Proactive Systems, Guildford, United Kingdom, 2007.
[10] R. Prasad (Ed.), My Personal Adaptive Global Net (MAGNET): Signals and Communication Technology, SpringerScience+Business Media, 2010. doi:10.1007/978-90-481-3437-3.
[11] http://www.ist-spice.org/.
[12] http://www.ist-e-sense.org/.
[13] http://www.sensei-project.eu/.
[14] Z. Abid, S. Chabridon, D. Conan, A Framework for Quality of Context Management, QuaCon 2009, LNCS 5786, Lecture notes published by Springer-Verlag Berlin Heidelberg, 2009, pp. 120–131.
[16] S. Blake et al., An Architecture for Differentiated Services, RFC 2475, IETF, December 1998.
[17] Martin Bogsted, Rasmus L. Olsen, Hans-Peter Schwefel, Probabilistic models for access strategies to dynamic information elements, Performance Evaluation 67 (2010) 43–60.
[18] H.P. Schwefel, M.B. Hansen, R.L. Olsen, Adaptive caching strategies for context management systems, in: Proceedings of PIMRC'07, Athens, Greece, 2007.
[20] Alexander Graham, Kronecker Products and Matrix Calculus with Applications, Ellis Horwood, 1981.
[21] Marcel Neuts, Structured Stochastic Matrices of M/G/1 Type and their Application, Dekker, New York, 1989.
[22] Lester Lipsky, Queueing Theory, A Linear Algebraic Approach, second ed., Springer, 2008.
[23] A. Corradi, M. Fanelli, L. Foschini, Adaptive context data distribution with guaranteed quality for mobile environments, Wireless Pervasive Computing (ISWPC) (2010).
[24] J.B. Filho, N. Agoulmine, A quality-aware approach for resolving context conflicts in context-aware systems, Embedded and Ubiquitous Computing (EUC) (2011).
[25] A. Toninelli, A. Corradi, R. Montanari, A quality of context-aware approach to access control in pervasive environments mobilewireless middleware, Operating Systems, and Applications (2009).

Further Reading

[15] K. Henricksen, J. Indulska, A. Rakotonirainy, Generating context management infrastructure from high-level context models, in: Proceedings of 4th International Conference on Mobile Data Management, 2003.
[19] A. Shawky, R. Olsen, J. Pedersen, J.G. Rasmussen, Service degradation in context management frameworks, in: 2011 Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN), July 31 2011–August 4, 2011, , pp. 1–6. http://dx.doi.org/10.1109/ICCCN.2011.6006060.

Ahmed Shawky received his bachelor degree in Computer Engineering from 6th of October University, Cairo, Egypt, in 2006. He received his master degree from Aalborg University in 2009 and is currently a PhD. Student in the Department of Electronic Systems at Aalborg University, Denmark. His research interests are in the areas of wireless networks, mobile networks, and network planning.

Rasmus Løvenstein Olsen is an Associate Professor at Aalborg University working in the Network and security group. Rasmus received his master degree from Aalborg University in 2003 and has received his PhD degree on the topic of Context Sensitive Service Discovery and Context Management with focus on access to dynamic information in 2008. Since, Rasmus have been teaching, supervising and working in various European projects and have more than 40 publications in papers for international conferences, journals and book chapters. Rasmus' current research focus is on service migration based on context information, context quality metrics and quality of service in context management systems, as well as usage of dynamic information elements in smart grid scenarios. Previous Rasmus has previous been guest visitor at National Institute of Communication Technology in Yokosuka Research Park in Japan, cooperating with a team of researchers on next generation network technology. He is currently engaged in European research projects doing research and leader of work packages.

Jens Pedersen is Associate Professor and head of the Networking and Security Section, Department of Electronic Systems, Aalborg University. His current research interests include network planning, traffic monitoring, and network security. He obtained his MSc in Mathematics and Computer Science from Aalborg University in 2002, and his PhD in Electrical Engineering also from Aalborg University in 2005. He is author/co-author of more than 70 publications in international conferences and journals, and has participated in Danish, Nordic and European funded research projects. He is also board member of a number of companies within technology and innovation.

Hans-Peter Schwefel is Scientific Director of the Forschungszentrum Telekommunikation Wien (FTW), Austria; he defines and coordinates FTW's overall research program on future communication technologies and their application in Telecommunications, Intelligent Transport Systems, and Intelligent Energy Distribution Grids. As a second affiliation, Hans is part-time Professor for IP-based wireless networks at Aalborg University, Denmark. Before Hans joint FTW in 2007, he has been full-time Associate Professor at Aalborg University since 2003, and prior to that Project Manager for Research Cooperations at Siemens Mobile Networks, Germany. Hans has been overall coordinator of the European FP6 project Highly-dependable IP-based Networks and Services (HIDENETS) and has had a leading role in several European projects. He is Associate Editor of IEEE Communications Letters and has been TPC co-chair of the International Symposium on Reliable Distributed Systems (SRDS) 2013, IEEE SmartGridComm Communications and Networks for Smart Grids and Smart Metering Symposium 2012, European Wireless 2011 and IEEE Network Computing and Applications 2010, and TPC member in many other relevant conferences. Hans obtained his doctoral degree (Dr. rer. nat.) in the area of IP traffic and performance modelling from the Technical University in Munich, Germany, in 2000.